University at Buffalo
Department of Computer Science and and Engineering
CSE 573 - Computer Vision and Image Processing

Fall 2023

Home Work #1
Due Date: 9/27/23, 11:59PM

1 Instructions

- Please submit a zip file with the name `UBID_hw1.zip` on UBlearns. UBID is the one that contains some combination of your name.
- `UBID_hw1.zip` contains two folders. One for section 2 with the title 'Coding' and another for section 3 with the title 'Written'.
- Follow the sub-folder structure mentioned in section 2 for the 'Coding' folder.
- For the 'Written' folder, export or scan your homework and store it as a PDF version before submitting online to UBLearns.

2 Coding: Image Manipulation (50 points)

The goal of this homework is to familiarize you with the OpenCV library and also conduct various image manipulation tasks. **We will be using OpenCV-Python version>=4.0 for all our homework in this course.**

Given a 512×256 RGB image (that you may download from the internet), your program should be able to conduct the following operations and output a resultant image.

- Gray scaling an image: Convert the original image to gray scale and save the output as `gray_image.png` (5 points).
- Scaling the gray-scale image: Scale the gray-scale image to half the size in both height and width *i.e.,* 256×128 and save the output as `gray_image_scaled.png` (5 points).

- Translating the gray-scale image: Translate the gray-scale image by 50 pixels to the right and 50 pixels to the bottom and save the output as `gray_image_translated.png` (5 points).
- Flipping the gray-scale image along horizontal axis: Flip the gray-scale image along the horizontal axis passing through the center of the image and save the output as `gray_image_flip_horizontal.png` (5 points).
- Flipping the gray-scale image along vertical axis: Flip the gray-scale image along the vertical axis passing through the center of the image and save the output as `gray_image_flip_vertical.png` (5 points).
- Invert gray-scale image: Conduct the gray-scale image inversion (i.e., 255 - gray_value or 1 - gray_value) and save the output as `gray_image_inversion.png` (5 points).
- Rotating the gray-scale image: Rotate the gray-scale image by 45 degrees in the clock-wise direction and save the output as `gray_image_rotated.png` (20 points).
- Also, save the original image as `image.png`.
- Bonus (10 points): You will receive a bonus of 10 points if you conduct scaling, translation and flipping an image along both horizontal and vertical axis on the original RGB image.

2.1 OpenCV Libraries permitted and prohibited

- You may only use OpenCV APIs for reading and writing an image. All other manipulations of the image have to be manually coded in python. DO NOT use 'cv2.cvtColor' or any other color space conversions.

2.2 Numpy Libraries permitted and prohibited

- You may only use the following numpy operations.

  Element-wise addition, subtraction, multiplication, and division

  Array slicing and indexing

  Reshaping and transposing

  Concatenation

  Matrix operations

  Generating array

  Boolean operation

  Element-wise comparison

2.3 Submission Folder Structure

Please submit the code written in python as `manipulation.py`.

- `UBID_hw1.zip`
  - `Coding`
    - `image.png`
    - `gray_image.png`
    - `gray_image_scaled.png`
    - `gray_image_rotated.png`
    - `gray_image_translated.png`
    - `gray_image_flip_horizontal.png`
    - `gray_image_flip_vertical.png`
    - `gray_image_inversion.png`
    - `image_scaled.png` (for bonus)
    - `image_translated.png` (for bonus)
    - `image_flip_horizontal.png` (for bonus)
    - `image_flip_vertical.png` (for bonus)
    - `manipulation.py`

2.4 Installation

- Remove installed opencv (Skip this step if you don't have it installed)

```
$ pip uninstall opencv-python
```

- Install opencv 4.6.0.66

```
$ pip install opencv-python==4.6.0.66
```

- Install opencv contribution library 4.6.0.66

```
$ pip install opencv-contrib-python==4.6.0.66
```

- An OpenCV installation guide is uploaded to piazza/resources

2.5 OpenCV Tutorial links.

- OpenCV-python documentation
- https://www.geeksforgeeks.org/opencv-python-tutorial/?ref=lbp
- https://www.mygreatlearning.com/blog/opencv-tutorial-in-python
- https://learnopencv.com/getting-started-with-opencv/
- https://pyimagesearch.com/2018/07/19/opencv-tutorial-a-guide-to-learn-opencv/

3 Written

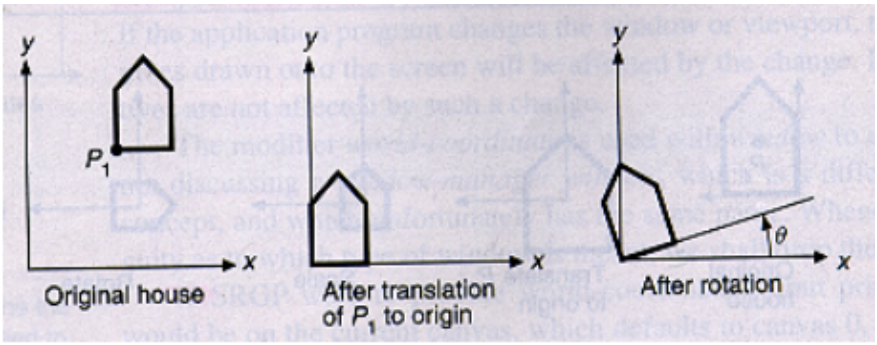
3.1 2D Transformations (15 points)

There are two steps in the house transformation shown below to map a point (x, y) to (x', y'): translation, and rotation.

The transformation between the first and the third 2D coordinates is

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = R \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \text{ where } R = R_{rotation} \cdot R_{translation}.$$

Suppose $P_1 = (10, 20)$ in the first coordinate and $\theta = 30^o$, calculate the transformation matrices $R_{translation}$, $R_{rotation}$, and R. (5 points/each)



3.2 3D Rotation (30 points)

Suppose we have a coordinate system A that can be mapped to a coordinate system B in two steps: 1) R_1: rotate π around X axes; 2) R_2: rotate $\pi/2$ around Z axes. The rotation follows right hand rule. The overall rotation R carries out this mapping from A to B.

- Give each 3x3 matrix R_1, R_2, R. (15 points, 5 points/each)
- Given a point whose coordinate is $[10, 0, 20]$ in A, calculate its coordinates B. (5 points)
- Give the 3x3 matrix R' carries out mapping from B to A. (5 points)
- Given a point whose coordinate is $[10, 0, 20]$ in B, calculate its coordinates A. (5 points)

3.3 Focal Length (5 points)

Given a lens, suppose the distance between an object and the lens is 6, the distance between the image plane and the lens is 3. In order to make the object in focus, please calculate the focal length f for the camera.

3.4 Submission Folder Structure

- `UBID_hw1.zip`

  `Written`

  `written.pdf`

4 Submission Guidelines

- Unlimited number of submissions is allowed and only the latest submission will be used for grading. Create `UBID_hw1.zip` and upload it to UBlearns.
- Identical code will be treated as plagiarism. Please work it out independently. Using online code or manipulated images available online will be considered as plagiarism.
- For code raising "RuntimeError", the grade will be ZERO for the homework.
- Any variations to the above submission folder structure will result in a ZERO for the homework. Also, there is no need to use any hard-coded local paths in your homework. Usage of such paths, would break when we run your code and will result in a ZERO.
- Late submissions guidelines apply for this homework.